



Navigation and Ancillary Information Facility

Introduction to the Family of SPICE Toolkits

October 2022



Topics

Navigation and Ancillary Information Facility

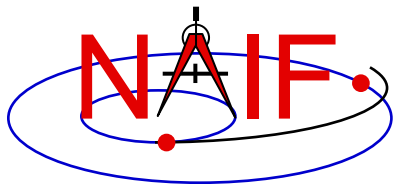
- **Architecture**
- **Contents**
- **Characteristics**
- **Versions**
- **Capabilities**
- **Directory Structure**
- **Application Programs**
- **Utility Programs**
- **Documentation**
- **Backup: Currently Supported Environments**



Toolkit Architecture

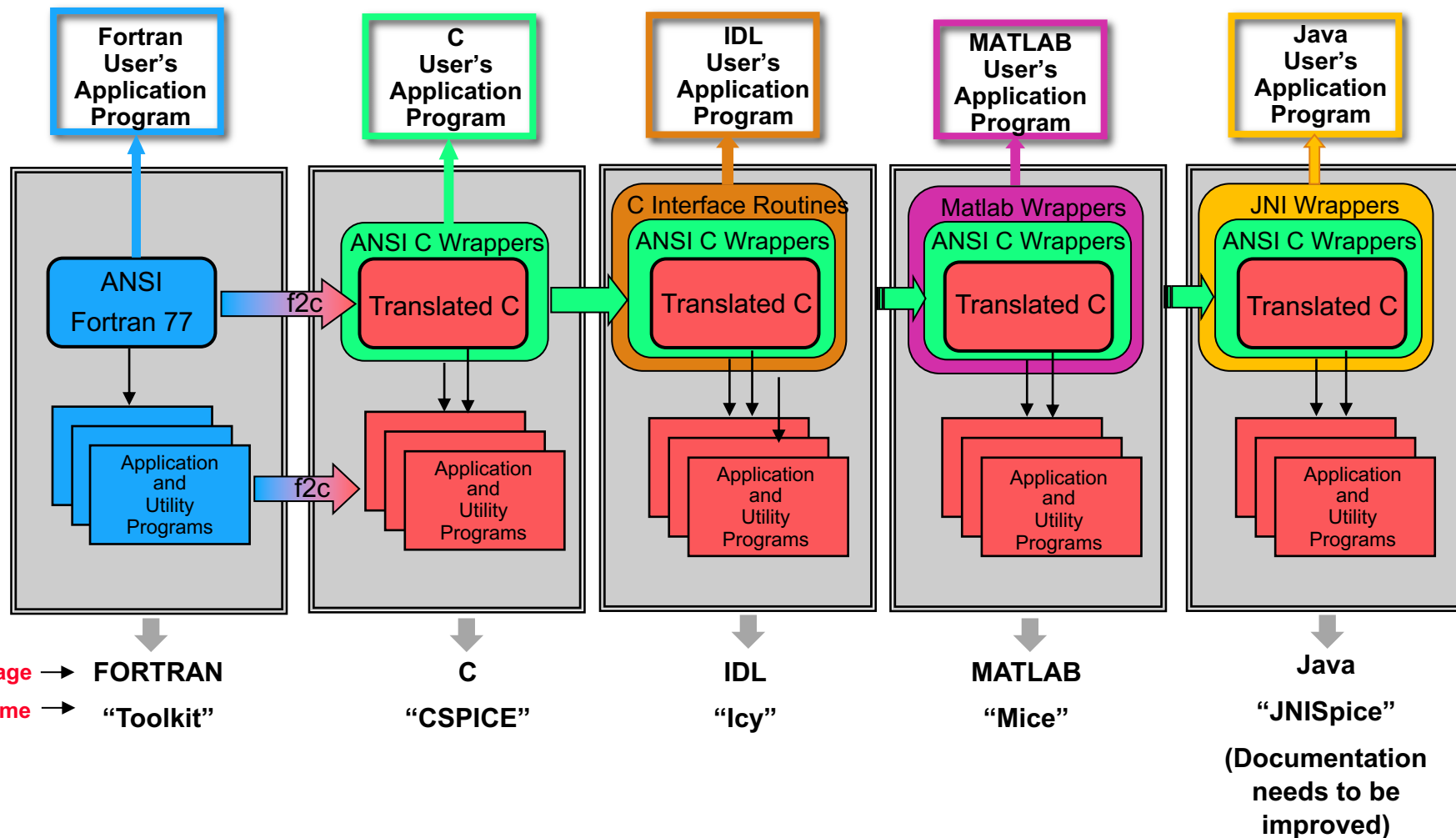
Navigation and Ancillary Information Facility

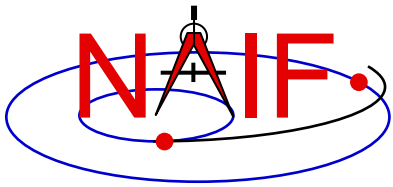
- **The SPICE Toolkit is officially available in Fortran 77, C, IDL and MATLAB.**
 - <https://naif.jpl.nasa.gov/naif/toolkit.html>
 - A beta Java Native Interface version (JNISpice) is also available
 - » <https://naif.jpl.nasa.gov/pub/naif/misc/JNISpice/>
- **The Toolkits are packaged and delivered as standalone products.**
 - The IDL, MATLAB and JNISpice Toolkits by necessity also include the complete C Toolkit.
- **Other people have created Python, Ruby, Swift and Julia toolkits, available from their own websites.**
 - <https://naif.jpl.nasa.gov/naif/links.html>
 - NAIF has NOT been involved in creating, testing or documenting these. Check with their authors about functionality and details.



Toolkit Architecture Pictorial

Navigation and Ancillary Information Facility





Fortran Toolkit

Navigation and Ancillary Information Facility

- **“Toolkit,” the Fortran 77 Toolkit.**
 - **Developed first: in use since February 1990.**
 - **Contains code written in ANSI Standard Fortran 77.**
 - » **A few widely supported non-ANSI extensions are used, for example `DO WHILE, DO...END DO`.**
 - **Compiles under a wide variety of Fortran compilers.**
 - » **While NAIF cannot guarantee proper functioning of SPICE under F90/F95 compilers except on officially supported environments, those compilers might properly compile SPICELIB with the resulting libraries being callable from F90/F95 code if that compiler supports the F77 standard.**

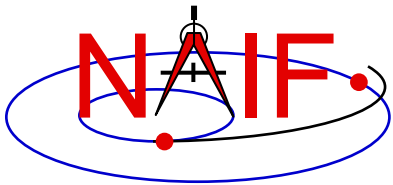


C Toolkit

Navigation and Ancillary Information Facility

- **“CSPICE,” the C-language Toolkit**
 - Designed to duplicate the functionality of the Fortran Toolkit.
 - All CSPICE source code is in ANSI C.
 - » The Fortran SPICE Toolkit code is converted to ANSI C using the automatic translation program f2c.
 - » High-level functions have been hand-coded in C and documented in C style in order to provide a natural C-style API. These functions are called “wrappers.”
 - » Most wrappers encapsulate calls to C functions generated by f2c
 - The simpler wrappers do their work in-line to boost performance
 - » f2c'd functions may be called directly, but this is strongly discouraged since f2c'd functions emulate Fortran functionality:
 - Call by reference
 - Fortran-style array indexing
 - Fortran-style strings

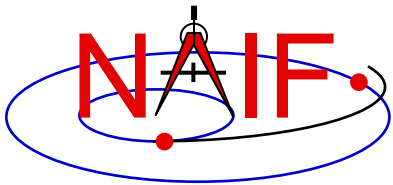
continued on next page



C Toolkit, continued

Navigation and Ancillary Information Facility

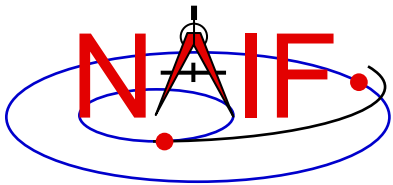
- **CSPICE runs under a wide variety of ANSI C compilers.**
- **CSPICE functions may be called from within C++ source code.**
 - » **CSPICE prototypes are protected from name mangling.**
- **Current CSPICE Limitations**
 - » **Not all “Required Reading” reference documents have been converted to C style, with C examples.**
 - **Eventually all will be converted.**
 - » **CSPICE wrappers do not exist for every API provided in the Fortran toolkits.**
 - **But CSPICE does include all the most commonly used modules.**
 - **More will be added as time permits.**
 - » **In some very limited cases, code generated by f2c fails to emulate Fortran accurately. Should not be a problem.**
 - **List-directed I/O has some problems (not consequential for CSPICE).**
 - **Treatment of white space in text output is slightly different in CSPICE.**
 - **Logical unit-to-file name translation does not handle file name “synonyms” properly under Linux: once opened with a specified name, a file must be referred to using the same name throughout a program run.**



IDL Toolkit

Navigation and Ancillary Information Facility

- **“Icy,” the Interactive Data Language Toolkit**
 - Provides an IDL-callable “wrapper” interface for many CSPICE wrapper routines.
 - » **Example:**
 - CSPICE: `spkezr_c (targ, et, ref, abcorr, obs, state, &itime);`
 - Icy: `cspice_spkezr, targ, et, ref, abcorr, obs, state, itime`
 - » NAIF will add additional interfaces to Icy as time permits.
 - By necessity all Icy Toolkit packages include the complete CSPICE Toolkit.
 - » **Additional Icy software components are:**
 - IDL interface wrappers (implemented in ANSI C)
 - Icy cookbook programs (implemented in IDL)
 - **Icy Documentation**
 - » **Icy Reference Guide**
 - Principal documentation showing how to call Icy wrappers.
 - Each Icy wrapper has an HTML page containing usage examples serving as the Icy “module header”.
 - » **Icy Required Reading**
 - Provides background information essential for programming with Icy.
 - **See the “IDL_Interface” tutorial for details**



Matlab Toolkit

Navigation and Ancillary Information Facility

- **“Mice,” the Matlab Toolkit**
 - Mice provides a Matlab-callable “wrapper” interface for many CSPICE wrapper routines
 - » **Example:**
 - CSPICE: `spkezc (targ, et, ref, abcorr, obs, state, &itime);`
 - Mice: `[state, ltime] = cspice_spkezc(targ, et, ref, abcorr, obs)`
 - By necessity all Mice Toolkit packages include the complete CSPICE Toolkit.
 - » **Additional Mice software components are:**
 - Matlab interface wrappers (implemented in Matlab wrapper scripts calling the ANSI C based interface library)
 - Mice cookbook programs (implemented in Matlab script)
 - **Mice Documentation**
 - » **Mice Reference Guide**
 - Principal documentation showing how to call Mice wrappers
 - Each Mice wrapper script has a documentation header containing usage examples, serving as SPICE “module header”, available from the `help` command. This documentation also exists as an HTML page.
 - » **Mice Required Reading**
 - Provides background information essential for programming with Mice
- **See the “Matlab_Interface” tutorial for details**



Toolkit Contents

Navigation and Ancillary Information Facility

- **Software**

- Subroutine libraries, with source code
 - » **SPICELIB (Fortran)**
 - » **CSPICE (C)**
 - » **Icy (C)**
 - » **Mice (C and Matlab scripts)**
- Executable programs
 - » **Application and utility programs**
 - » **A few example programs (called “cookbook” programs)**
- Installation/build scripts (normally you do NOT need to use these)

- **Documentation**

- Available in plain text and HTML

- **Example Data**

- Sample kernel files (supplied only for use with cookbook example programs, not valid for general use).



Toolkit Characteristics

Navigation and Ancillary Information Facility

- **Computations are identical in all languages.**
- **For a given computer and operating system, all Toolkits use identical kernel files.**
 - Refer to the “Porting Kernels” tutorial for information about using kernels received from a machine different from what you are using.
- **Code is well tested before being released to users.**
- **New Toolkits are always backwards compatible.**
 - An application that worked when linked against an older Toolkit will link and work, without need for changes, using a new Toolkit.
 - Past functionality is never changed or removed, except that:
 - » enhancements of existing routines are allowed.
 - » NAIF reserves the right to fix bugs.
- **Extensive user-oriented documentation is provided.**
 - Includes highly documented source code.



Toolkit Versions

Navigation and Ancillary Information Facility

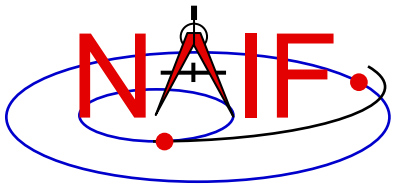
- **Toolkit Version**
 - **SPICE Toolkits have an associated Version number**
 - » Example: “N0067” (also written as “N67”)
 - **The version number applies to all language implementations for all supported platforms.**
- **When does NAIF release a new SPICE Toolkit version?**
 - » Not according to a fixed schedule
 - » Primarily driven by availability of significant new capabilities
 - For example, addition of the digital shape kernel subsystem (DSK)
 - » On rare occasion a Toolkit update is released to fix bugs, improve documentation, or satisfy an urgent request from a flight project.



Toolkit Library Overview

Navigation and Ancillary Information Facility

- **Toolkit libraries contain a broad set of capabilities related to the computations needed for determining “observation geometry” and time conversions.**
 - **Examples appear on the next several pages**
- **Not all functionality is present in all four language versions of the Toolkit library.**
 - **The Fortran (Toolkit) and C (CSPICE) Toolkits provide almost identical functionality.**
 - **The IDL (Icy) and Matlab (Mice) Toolkits duplicate most but not all of the functionality available in the C Toolkits.**
 - » **We add additional interfaces as time permits.**



Toolkit Library Capabilities - 1

Navigation and Ancillary Information Facility

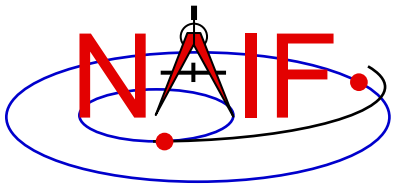
- **Kernel read access**
 - “Load” kernels
 - Get state or position vectors (SPK)
 - Get orientation of planets, natural satellites, etc. (PCK)
 - Get body shape parameters or physical constants (PCK)
 - Get orientation of spacecraft or spacecraft instruments or structures (CK, FK)
 - Get instrument parameters (e.g., FOV) (IK)
 - Get digital shape data (DSK)
 - Query binary EK files (EK-ESQ)
- **Kernel write access for binary kernels**
 - SPK writers
 - CK writers
 - PCK writers (only for binary PCK files)
 - DSK writers



Toolkit Library Capabilities - 2

Navigation and Ancillary Information Facility

- **Additional ephemeris functions**
 - Classical osculating elements
 - Two-body Keplerian propagation
 - NORAD two line elements sets (TLE) propagation
 - Light time and Stellar aberration computation
- **Frame transformations**
 - Obtain 3x3 matrices for frame transformations of positions
 - Obtain 6x6 matrices for frame transformations of states
- **Time conversions**
 - Conversion between standard systems: TDB, TT (TDT), UTC
 - Conversion between SCLK and other systems
 - Parsing and formatting
- **Geometry finder calculations**
 - Find times or time spans when a specified geometric situation is true
 - Find times or time spans when a specified geometric parameter is within a given range, or is at a maximum or minimum



Toolkit Library Capabilities - 3

Navigation and Ancillary Information Facility

- **Math**
 - Vector/Matrix operations
 - Rotations, Euler angles, quaternions
 - Coordinate conversion (systems: latitudinal, cylindrical, rectangular, RA and DEC, spherical, geodetic, planetographic)
 - Geometry: ellipsoids, ellipses, planes
 - High-level functions: illumination angles, sub-observer point, sub-solar point, surface intercept point.
- **Constants**
 - Julian date of epoch J2000, SPD (seconds per day), PI, etc.
- **Strings**
 - Parsing: find tokens, words
 - Numeric conversion
 - Pattern matching
 - Replace marker, substring
 - Suffix, prefix
 - Case conversion
 - Find first/last non-blank character, first/last printing character



Toolkit Library Capabilities - 4

Navigation and Ancillary Information Facility

- **Arrays**
 - Sorting, finding order vector, reordering
 - Searching: linear, binary
 - Insertion and deletion
- **Name/ID code conversion**
 - Bodies
 - Frames
 - Surfaces
- **I/O support**
 - Logical unit management (for Fortran toolkits)
 - Open, read, write text files
 - Kernel pool API
- **Exception handling**
 - Control exception handling behavior: mode, set message, assign output device.
- **Advanced data types**
 - Cells, Sets
 - Windows (sometimes called schedules)
 - Symbol Tables
 - Planes, Ellipses



Toolkit Directory Structure

Navigation and Ancillary Information Facility

- **The directory structures for the four kinds of Toolkits are almost identical. However...**
 - The CSPICE, Icy and Mice Toolkits also have a directory for include files.
 - The names for application source code directories in CSPICE, Icy and Mice differ slightly from those in the Fortran toolkit.
 - Icy and Mice include additional directories for :
 - » Icy/Mice source code
 - » Icy/Mice cookbook programs
- **The top level directory name for each Toolkit is:**
 - “toolkit” for Fortran Toolkits.
 - “cspice” for C Toolkits.
 - “icy” for IDL Toolkits.
 - “mice” for Matlab Toolkits
 - “JNISpice” for Java Toolkits



Toolkit Directory Structure

Navigation and Ancillary Information Facility

- **The next level is comprised of:**
 - **data**
 - » Cookbook example kernels (use **ONLY** for training with cookbook programs).
 - **doc**
 - » Text documents — *.req, *.ug, spicelib.idx/cspice.idx, whats.new, dscripnt.txt, version.txt.
 - » Subdirectory containing HTML documentation, called “html”.
 - The “html” subdirectory contains a single file — the top level HTML documentation index called “index.html” — and a number of subdirectories, one for each of the various groups of documents in HTML format (API Reference Guide pages, User’s Guide pages, etc.).
 - **etc**
 - » In most Toolkits this directory is empty.
 - **exe**
 - » Executables for some SPICE application and utility programs:
 - brief, chronos, ckbrief, commnt, dskbief, dskeyp, frmdiff, inspekt, mkdsk, mkspk, msopck, spact, spkdiff, spkmerge, tobin, toxfr, version.
 - » Executables for the several cookbook example programs:
 - simple, states, subpt, tictoc



Toolkit Directory Structure

Navigation and Ancillary Information Facility

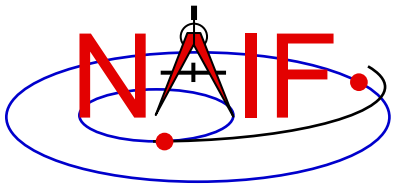
- **include** (applies only to CSPICE, Icy, Mice, and JNISpice)
 - » **API header files.**
 - File to include in callers of CSPICE is SpiceUsr.h
- **lib**
 - » **Toolkit libraries:**
 - **For Fortran Toolkits**
 - spicelib.a or spicelib.lib (public modules; use these)
 - support.a or support.lib (supporting modules; don't use these)
 - **For C Toolkits**
 - cspice.a or cspice.lib (public modules; use these)
 - csupport.a or csupport.lib (supporting modules; don't use these)
 - **For Icy Toolkits:**
 - icy.so or icy.dll (shared object library)
 - icy.dlm (dynamically loadable module)
 - cspice.a or cspice.lib, and csupport.a or csupport.lib
 - **For Mice Toolkits:**
 - mice.mex* (shared object library)
 - cspice.a or cspice.lib, and csupport.a or csupport.lib
 - **For JNISpice Toolkits:**
 - libJNISpice.so or libJNISpice.lib (shared object library)
 - cspice.a or cspice.lib, and csupport.a or csupport.lib
- **src**
 - » **Source code directories for executables and libraries**
 - Files have type *.f, *.for, *.inc, *.pgm, *.c, *.h, *.x, *.pro, *.m, *.java, *.class, *.html
 - *.h files appearing here are not part of the user API



Toolkit Application Programs

Navigation and Ancillary Information Facility

- **SPICE Toolkit application programs are available to:**
 - create most binary kernel types
 - compare or analyze certain kernel types
 - do various kinds of time conversions
 - and more...
 - See the toolkit_apps tutorial for details
- **Some additional application programs are available only from the NAIF website:**
 - <http://naif.jpl.nasa.gov/naif/utilities.html>
 - See the non_toolkit_apps tutorial for details



Toolkit Utility Programs

Navigation and Ancillary Information Facility

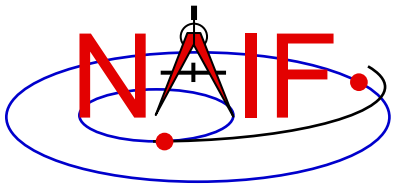
- **SPICE Toolkit utility programs are available to:**
 - **add comments to binary kernels**
 - » `commnt`
 - **read comments from binary kernels**
 - » `commnt`, `spacit`
 - » `inspekt` (only for EK/ESQ files)
 - **summarize coverage of binary kernels**
 - » `brief`, `ckbrief`, `dskbrief`, `spacit`
 - **merge or subset SPK files**
 - » `spkmerge`
 - **indicate current Toolkit version**
 - » `version`
 - **port binary SPICE kernels between incompatible systems (infrequently needed)**
 - » `tobin`, `toxfr`, `spacit`
 - » `bingo` (available only from the NAIF webpage)
 - **port text SPICE kernels between incompatible systems**
 - » `bingo` (available only from the NAIF webpage)
- **See the `toolkit_apps` tutorial for details**



Toolkit Documentation - 1

Navigation and Ancillary Information Facility

- **All Toolkits include documentation in plain text and HTML formats.**
 - Plain text documents are located under the “doc” directory
 - HTML documents are located under the “<toolkit_name>/doc/html” (Unix) or “<toolkit name>\doc\html” (Windows) directory
 - » “index.html” is the top level index... your starting point
- **All Toolkits include the following kinds of documents**
 - **Module headers**
 - » They act as primary functional specification: I/O, exceptions, particulars defining behavior of module
 - » They contain code examples
 - » A standard format is used for each routine or entry point
 - » **Location of HTML Module Headers:**
 - Use the “API Reference Guide” link from the top level index
 - » **Location of plain text Module Headers:**
 - Fortran: the top comment block in the source code files under “src/spicelib”
 - C: the top comment block in the source code files under “src/cspice”
 - IDL: Icy Module Headers are not available in plain text format
 - Matlab: accessible via “help *function_name*” command



Toolkit Documentation - 2

Navigation and Ancillary Information Facility

- **“Required Reading” documents**
 - » **Extensive technical references for principal subsystems**
 - Provide many low-level details
 - Provide code examples
 - » **HTML versions are accessible using the “Required Reading Documents” link from the top level index.**
 - » **Plain text versions are located under “doc” and have extension “.req”**
 - » **Not all Required Readings were adapted for all languages**
 - Some of the Required Reading documents provided with CSPICE are based upon Fortran SPICE
 - Some of the Required Readings for Icy or Mice toolkits are based upon CSPICE
- **User’s Guides**
 - » **Tell how to use the utility and application programs.**
 - » **HTML versions are accessible using the “User’s Guide Documents” link from the top level index.**
 - » **Plain text versions are located under “doc” and have extension “.ug.”**



Toolkit Documentation - 3

Navigation and Ancillary Information Facility

- **Other documents**

- **Permuted Index**

- » **Maps phrases describing functionality to corresponding module names and file names**
 - » **Shows names of all entry points in Fortran toolkit APIs**
 - » **HTML version is accessible using the “Permuted Index” link from the top level index.**
 - » **Plain text version is located under “doc” and has extension “.idx”:**
 - **Fortran:** spicelib.idx
 - **C:** cspice.idx
 - **IDL:** icy.idx and cspice.idx
 - **Matlab:** mice.idx and cspice.idx

- **Toolkit Description**

- » **Describes the directory structure and contents of an installed Toolkit**
 - » **Customized based on set of delivered products and platform**
 - » **HTML version is accessible using the “Toolkit Contents” link from the top level index.**
 - » **Plain text version is “doc/dscriptn.txt”**

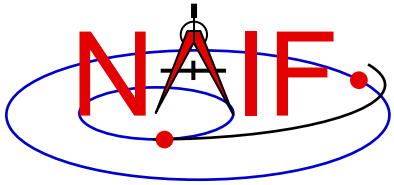
continues on next page



Toolkit Documentation - 4

Navigation and Ancillary Information Facility

- **Introduction to SPICE**
 - » **HTML document containing a brief introduction to the Toolkit and SPICE system; accessible using the “Introduction to the SPICE System” link from the top level index.**
- **What’s New in SPICE**
 - » **Describes new features and bug fixes in each Toolkit release, covering the last 20 years.**
 - » **Plain text version is “doc/whats.new”.**
 - » **HTML version is accessible using the “What’s New in SPICE” link from the top level index.**
- **Toolkit Version Description**
 - » **Indicates Toolkit version**
 - » **Plain text version is “doc/version.txt”**
 - » **Not available in HTML**
- **You can also use Google to view any of the Toolkit documents**
 - **E.g., search on the API name (e.g. spkezr, spkezr_c, or cspice_spkezr (for Icy and Mice) in combination with naif and spice**
 - **E.g. search on the document name (e.g. CK Required Reading, spkdiff user’s guide)**



Navigation and Ancillary Information Facility

Backup

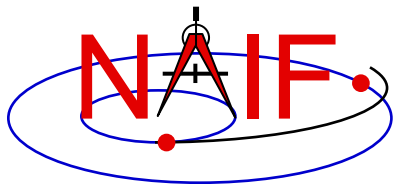
Supported Environments



Supported Environments

Navigation and Ancillary Information Facility

- **NAIF ports the SPICE Toolkit to many popular environments.**
 - Each environment is characterized by
 - » Language
 - » Hardware type (platform)
 - » Operating System
 - » Compiler
 - » Selected compilation options
- **NAIF provides SPICE Toolkit packages for each supported environment.**
 - If you cannot find a package built for the environment of interest to you, contact NAIF.
- **The list of supported environments slowly evolves.**
 - Old ones no longer supportable are terminated.
 - New ones are added based on user interest and available NAIF resources.

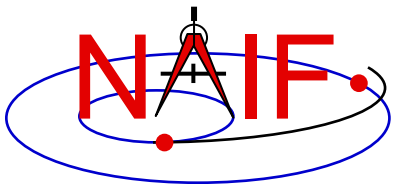


Supported Environments – Fortran – N0067

Navigation and Ancillary Information Facility

Product Name	Operating System	Compiler
Mac/Intel, OS-X, Intel FORTRAN, 64bit	OS X 10.15	Intel Fortran 2021.4.0
Mac/Intel, OS-X, gfortran, 64bit	OS X 10.15	gfortran 9.1
PC, CYGWIN, gfortran, 64bit	Windows/Cygwin 10.0	gfortran 11.2
PC, Linux, Intel FORTRAN, 32bit	Red Hat Linux (RHE7)	Intel Fortran 13.1
PC, Linux, Intel FORTRAN, 64bit	Red Hat Linux (RHE7)	Intel Fortran 13.1
PC, Linux, g77, 32bit	Red Hat Linux (RHE6)	g77 3.4
PC, Linux, gfortran, 32bit	Red Hat Linux (RHE7)	gfortran 7.3
PC, Linux, gfortran, 64bit	Red Hat Linux (RHE7)	gfortran 7.3
PC, Windows, Intel FORTRAN, 32bit	Windows 10	Intel Fortran 2021.4.0
PC, Windows, Intel FORTRAN, 64bit	Windows 10	Intel Fortran 2021.4.0
Sun/SPARC, Solaris, SUN FORTRAN, 32bit	Solaris 10	Sun FORTRAN 95 8.8

The OS and compiler versions listed above were used to build and test N0067 Fortran SPICE Toolkit distribution packages. These packages are expected to work on/with newer OS and compiler versions.



Supported Environments – C – N0067

Navigation and Ancillary Information Facility

Product Name	Operating System	Compiler
Mac/Intel, OS-X, Apple C, 32bit	OS X 10.15	Apple C 12.0
Mac/M1, OS-X, Apple C, 64bit	OS X 12.0	Apple C 13.0
PC, CYGWIN, gCC, 64bit	Windows/Cygwin 10.0	gcc 11.2
PC, Linux, gCC, 32bit	Red Hat Linux (RHE7)	gcc 7.3
PC, Linux, gCC, 64bit	Red Hat Linux (RHE7)	gcc 7.3
PC, Windows, Microsoft Visual C, 32bit	Windows 10	MS Visual Studio 16.11 C
PC, Windows, Microsoft Visual C, 64bit	Windows 10	MS Visual Studio 16.11 C
Sun/SPARC, Solaris, gCC, 32bit	Solaris 10	gcc 3.4
Sun/SPARC, Solaris, gCC, 64bit	Solaris 10	gcc 3.4
Sun/SPARC, Solaris, SUN C, 32bit	Solaris 10	Sun C 5.15
Sun/SPARC, Solaris, SUN C, 64bit	Solaris 10	Sun C 5.15

The OS and compiler versions listed above were used to build and test N0067 CSPICE Toolkit distribution packages. These packages are expected to work on/with newer OS and compiler versions.

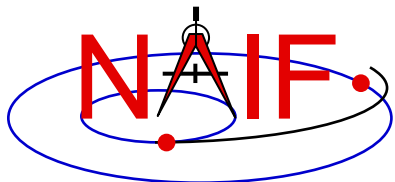


Supported Environments - IDL

Navigation and Ancillary Information Facility

Product Name	Operating System	Compiler, IDL
Mac/Intel, OS-X, Apple C/IDL, 64bit	OS X 10.15	Apple C 12.0, IDL 8.7
PC, Linux, gcc/IDL, 64bit	Red Hat Linux (RHE7)	gcc 7.3, IDL 8.8
PC, Windows, Microsoft Visual C/IDL, 64bit	Windows 10	MS Visual Studio 16.11 C, IDL 8.8

The OS, compiler, and IDL versions listed above were used to build and test N0067 Icy Toolkit distribution packages. These packages are expected to work on/with newer OS, compiler, and IDL versions.



Supported Environments - MATLAB

Navigation and Ancillary Information Facility

Product Name	Operating System	Compiler, MATLAB
Mac/Intel, OS-X, Apple C, 64bit	OS X 10.15	Apple C 12.0.0, MATLAB R2018b
PC, Linux, gCC, 64bit	Red Hat Linux (RHE7)	gcc 7.3, MATLAB R2020b
PC, Windows, Microsoft Visual C/Matlab, 64bit	Windows 10	MS Visual Studio 16.11 C, MATLAB R2020b

The OS, compiler, and Matlab versions listed above were used to build and test N0067 MiceToolkit distribution packages. These packages are expected to work on/with newer OS, compiler, and Matlab versions.



Status for Other Environments

Navigation and Ancillary Information Facility

- **The SPICE and CSPICE packages should function as expected on platforms running any Linux OS (Ubuntu, Fedora, etc.), BSD OS (OpenBSD, FreeBSD, etc.), or a Linux based OS environment (minGW) using a standard GCC tool-chain (gfortran or gcc compiler).**
 - Version 4.2 or later for gfortran; 4.0 or later for gcc
- **The Mice package has been successfully built against the octave environment (version > 3.4) on Linux and OS X. Contact NAIF if you have questions concerning use with Octave.**